Beehive Controls Tutorial

Table of contents

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

1. Introduction

1.1. Tutorial Goals

In this tutorial, you will learn:

- what Beehive Controls are for and what they do.
- how to create a Beehive Control interface and implementation.
- how to compile a Beehive Control.
- how to use a Beehive Control as a component in a larger application.
- how to use metadata annotations in Beehive Controls.
- how to create custom metadata annotations.

2. Step 1: Begin the Control Tutorial

2.1. To Set up the Development Environment

Complete all of the necessary and optional steps in the following topic: Beehive Installation and Setup (../setup.html)

After completing the instructions, leave the command shell open to use throughout this tutorial.

Before proceding, confirm that you have the following variables set in your shell:

- `ANT_HOME`
- `JAVA_HOME`
- `CATALINA_HOME`

Also ensure that the following elements are on your `PATH`:

- `ANT_HOME/bin`
- `JAVA_HOME/bin`

2.2. To Create a Development and Test Application

In this step you create a web application that serves as the development and testing ground for your control. To test the control, you will invoke the control methods from the web application.

Copy the folder `<BeehiveRoot>/samples/netui-blank` into `C:/beehive_projects`.

Note:

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

> <BeehiveRoot> refers to the top-level directory of your Beehive installation. A typical value for <BeehiveRoot> is `C:/apache/apache-beehive-1.0`.

Rename the folder `C:/beehive_projects/`**`netui-blank`** to the name `C:/beehive_projects/`**`control_tutorial`**

Before proceeding, confirm that the following directory structure exists:

```
C:
  beehive_projects
    control_tutorial
      resources
      WEB-INF
      Controller.java
      index.jsp
```

2.3. Edit the build.properties File

Next you will edit the `build.properties` file--the file that sets the build-related properties for your test environment.

Open the file `C:/beehive_projects/control_tutorial/WEB-INF/src/build.properties` in a text editor.

Edit the `beehive.home` property so it points to the top-level folder of your beehive installation.

Edit the `context.path` property so it has the value **`control_tutorial`** (as shown below).

For example, if your beehive installation resides at `C:/apache/apache-beehive-1.0`, then your `build.properties` file would appear as follows.

```
beehive.home=C:/apache/apache-beehive-1.0

servlet-api.jar=${os.CATALINA_HOME}/common/lib/servlet-api.jar
jsp-api.jar=${os.CATALINA_HOME}/common/lib/jsp-api.jar

context.path=control_tutorial
```

> **Note:**
> Windows users must use forwardslashes (/) not backslashes (\) in the `build.properties` file.

2.4. Add the Control Interface and Implementation Files

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Copy the folder **<BeehiveRoot>/samples/controls-blank/src/pkg** into the folder **C:/beehive_projects/control_tutorial/WEB-INF/src**.

Before proceeding, confirm that the following directory structure exists:

```
C:
  beehive_projects
    control_tutorial
      WEB-INF
        src
          pkg
            Hello.java
            HelloImpl.java
```

2.5. To Start the Tomcat Server

At the command prompt, enter:

```
%CATALINA_HOME%\bin\startup.bat
```

3. Step 2: Compile the Control Implementation and Interface Files

3.1. Introduction

A Beehive Control consists of two files: an interface file and an implementation file. The interface file is the public face of your control. It lists all of the methods which can be invoked by users. The implementation file contains the implementation code for the methods listed in the interface file.

3.2. To Examine the Control Files

Open the file C:/beehive_projects/control_tutorial/WEB-INF/src/pkg/HelloImpl.java.

The implementation file appears as follows. (There is no need to edit the file at this point in the tutorial.)

```
package pkg;

import org.apache.beehive.controls.api.bean.*;

@ControlImplementation(isTransient=true)
public class HelloImpl implements Hello
{
    public String hello()
    {
        return "hello!";
    }
}
```

© 2005, Apache Software Foundation

Copyright © 2004 The Apache Software Foundation. All rights reserved.

Open the file `C:/beehive_projects/control_tutorial/WEB-INF/src/pkg/Hello.java`.

The interface file appears as follows. (There is no need to edit the file at this point in the tutorial.)

```
package pkg;

import org.apache.beehive.controls.api.bean.*;

@ControlInterface
public interface Hello
{
    String hello();
}
```

3.3. Edit the Controller.java File

To test the Hello control, you need to call the control from some other resource, such as a JAVA application, JSP page, a web application, etc. In the following two steps you will call the control from the Contoller class in a web application and display the results on a JSP page.

Open the file `C:/beehive_projects/control_tutorial/Controller.java` in a text editor.

Edit `Controller.java` so it appears as follows:

```
import javax.servlet.http.HttpSession;

import org.apache.beehive.netui.pageflow.Forward;
import org.apache.beehive.netui.pageflow.PageFlowController;
import org.apache.beehive.netui.pageflow.annotations.Jpf;

import org.apache.beehive.controls.api.bean.Control;
import pkg.Hello;

@Jpf.Controller(
    simpleActions={
        @Jpf.SimpleAction(name="old_begin", path="index.jsp")
    },
    sharedFlowRefs={
        @Jpf.SharedFlowRef(name="shared", type=shared.SharedFlow.class)
    }
)
public class Controller
    extends PageFlowController
{
    @Jpf.SharedFlowField(name="shared")
    private shared.SharedFlow sharedFlow;
```

© 2005, Apache Software Foundation

Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
    @Control
    private Hello _helloControl;

    @Jpf.Action(
      forwards={
        @Jpf.Forward(name="success", path="index.jsp")
      }
    )
    protected Forward begin() throws Exception
    {
        Forward f = new Forward("success");
        f.addActionOutput("helloMessage", _helloControl.hello());
        return f;
    }

    /**
     * Callback that is invoked when this controller instance is created.
     */
    protected void onCreate()
    {
    }

    /**
     * Callback that is invoked when this controller instance is destroyed.
     */
    protected void onDestroy(HttpSession session)
    {
    }
}
```

The two import statements import the @Control annotation and the Hello control, respectively.

The `@Jpf.SimpleAction` named "`begin`" is renamed to "`old_begin`" because the Beehive runtime automatically looks for and runs any action named "`begin`" when a Page Flow is first instantiated. The renaming makes the runtime run the action method `begin()`, which allows us to call the Hello control inside of the method body.

3.4. To Edit the index.jspTest Page

Edit the file `C:/beehive_projects/control_tutorial/index.jsp` so it appears as follows. Code to edit appears in bold.

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-template-1.0"
prefix="netui-template"%>
<netui:html>
```

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

```
  <head>
    <title>Control Tutorial Test Page</title>
    <netui:base/>
  </head>
  <netui:body>
    <h3>Control Tutorial Test Page</h3>
    <p>
        Response from the hello() method on the Hello Control:
        <netui:span style="color:#FF0000"
value="${pageInput.helloMessage}"/>
    </p>
  </netui:body>
</netui:html>
```

3.5. To Compile and Deploy the Control

You are now ready to compile the test web application and the embedded the control.

To build the web application, enter:

```
ant
  -f C:\beehive_projects\control_tutorial\WEB-INF\src\build.xml
  clean
  build
  war

Copy and Paste version:
ant -f C:\beehive_projects\control_tutorial\WEB-INF\src\build.xml clean
build war
```

To deploy the web application to Tomcat, copy the WAR file into Tomcat's `webapps/` directory.

```
copy C:\beehive_projects\control_tutorial.war %CATALINA_HOME%\webapps /Y
```

3.6. To Test the Control

Open a web browser and enter the following in the address bar:

```
  http://localhost:8080/control_tutorial/begin.do
```

You will be directed to the index.jsp page.

Note the message on the page: "hello!"

This message is provided by the Hello control.

4. Step 3: Add a Parameterized Method to the Control

4.1. To edit the Interface and Implementation Files

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Edit `C:/beehive_projects/control_tutorial/WEB-INF/src/pkg/HelloImpl.java` so it appears as follows. Code to add appears in bold.

```
package pkg;

import org.apache.beehive.controls.api.bean.*;

@ControlImplementation(isTransient=true)
public class HelloImpl implements Hello
{
    public String hello()
    {
        return "hello!";
    }

    public String helloParam( String name )
    {
        return "Hello, " + name + "!";
    }
}
```

Edit `C:/beehive_projects/control_tutorial/WEB-INF/src/pkg/Hello.java` so it appears as follows. Code to add appears in bold.

```
package pkg;

import org.apache.beehive.controls.api.bean.*;

@ControlInterface
public interface Hello
{
    String hello();

    String helloParam( String name );
}
```

4.2. Edit the Controller.java File

Edit the `begin()` method in `Controller.java` so it appears as follows. Code to add appears in bold.

```
...

public class Controller
    extends PageFlowController
{

    ...

    @Jpf.Action(
      forwards={
```

© 2005, Apache Software Foundation

Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
        @Jpf.Forward( name="success", path="index.jsp" )
      }
    )
    protected Forward begin() throws Exception
    {
        Forward f = new Forward("success");
        f.addActionOutput("helloMessage", _helloControl.hello());
        f.addActionOutput("helloParamMessage",
_helloControl.helloParam("World"));
        return f;
    }

   ...

}
```

4.3. To Edit the Test JSP Page

Edit `index.jsp` so it appears as follows. Code to add appears in bold.

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-template-1.0"
prefix="netui-template"%>
<netui:html>
  <head>
    <title>Control Tutorial Test Page</title>
    <netui:base/>
  </head>
  <netui:body>
    <jsp:useBean class="pkg.HelloBean" id="helloBean" scope="session"/>
    <h3>Control Tutorial Test Page</h3>
    <p>
        Response from the hello() method on the Hello Control:
        <netui:span style="color:#FF0000"
value="${pageInput.helloMessage}"/>
    </p>
    <p>
        Response from the helloParam() method on the Hello Control:
        <netui:span style="color:#FF0000"
value="${pageInput.helloParamMessage}"/>
    </p>
  </netui:body>
</netui:html>
```

4.4. To Compile and Redeploy the Control

Compile and deploy the web application using the same command line statements used in step 2.

© 2005, Apache Software Foundation

Copyright © 2004 The Apache Software Foundation. All rights reserved.

4.5. To Test the Control

Open a web browser and enter the following in the address bar:

```
http://localhost:8080/control_tutorial/begin.do
```

You will be directed to the index.jsp page.

Note the messages on the page: "hello!" and "Hello, World!"

These messages are provided by the Hello control.

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.
© 2005, Apache Software Foundation